WIP: The Optimization of Alcoholism under a Hypothetical Bartering System

Kevin Palani and Kevin Zheng

August 2, 2019

Contents

1 Introduction to the hypothetical bartering system **1**

2 Modeling amount of caps and bottles **1**
2.1 Vector representation of caps and bottles 1
2.2 Matrix representation of the bartering system 1

3 Computing the final state of bartering **3**
3.1 Algorithm 3
3.2 Algebraic Solution 4

4 Using the final state of the vector to deduce the amount of drinks one had **4**

1 Introduction to the hypothetical bartering system

You have \$10, and a beer is 2\$. Very quickly you can see that if you spend all 10\$, you will get 5 beers. Once you've drunken the 5 beers, you are left with 5 beer bottles and 5 caps. The store owner strikes you a deal. If you give him two empty bottles or four bottle caps, he'll give you a new bottle of beer. He is also kind enough to let you drink before you pay. How many drinks you can get?

2 Modeling amount of caps and bottles

2.1 Vector representation of caps and bottles

$$\begin{bmatrix} a \\ b \\ 1 \end{bmatrix}$$

Will be the vector that represents the bottles and caps such that a is the amount of bottles, and b is the amount of caps. The 1 is a homogenization of the vector.

2.2 Matrix representation of the bartering system

If we can spend two empty bottles and receive a full drink, that is equivalent to spending two bottles and getting one bottle and one cap. We will represent this operation as the following translational matrix.

$$B = \begin{bmatrix} 1 & 0 & -2+1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

And since we can drink before we pay, having only one empty bottle is enough to drink.

$$B = \begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

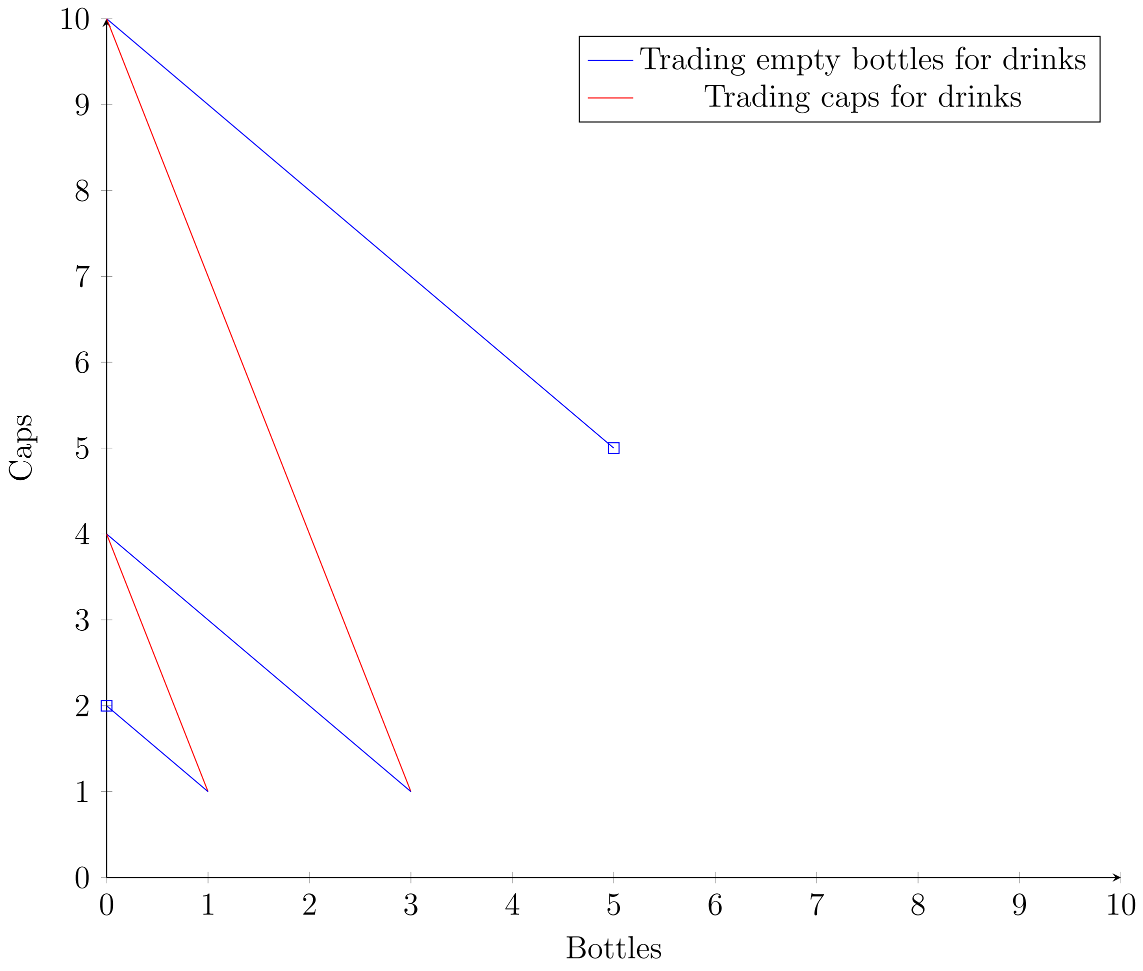
The purchasing of a full drink using 4 caps can be similarly represented as a translational matrix.

$$C = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -4+1 \\ 0 & 0 & 1 \end{bmatrix}$$

Which can simply be evaluated to.

$$C = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -3 \\ 0 & 0 & 1 \end{bmatrix}$$

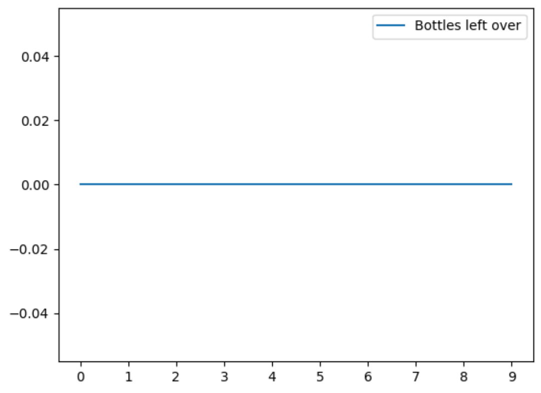
These two operations can be represented geometrically as a translation of a point on a 2 dimensional Cartesian plane. For example, if we start with 5 empty bottles and 5 empty caps, we can trace the motion of the point as following.

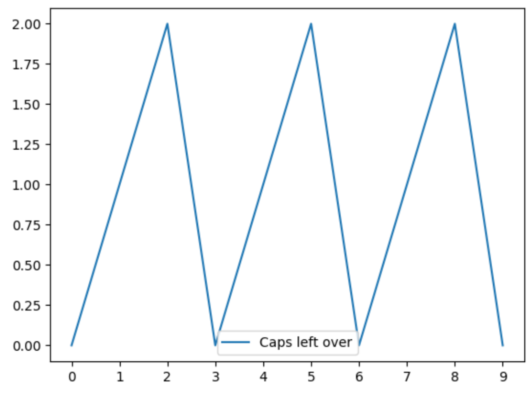


3 Computing the final state of bartering

3.1 Algorithm

```
#!/usr/bin/python
import matplotlib.pyplot as plt

cap = 4 - 1
bot = 2 - 1

capcount = 0
botcount = 0

def final(a, b):
    global capcount
    global botcount
    if a / cap > 0:
        capcount = capcount + (a / cap)
        return final(a % cap , b)
    elif b / bot > 0:
        botcount = botcount + (b / bot)
        return final(a, b % bot)
    return (a, b)

finalVectors = [final(i, i) for i in range(10)]

caps = [i for i,j in finalVectors]
bots = [j for i,j in finalVectors]

x = range(10)

plt.plot(x, bots, label="Bottles left over")
plt.xticks(x)
plt.legend()
plt.savefig("bots1.png")
plt.clf()
plt.plot(x, caps, label="Caps left over")
plt.xticks(x)
plt.legend()
plt.savefig("caps1.png")
```





From this it makes sense that you'd never end up with a bottle since having one left would always allow you to get another one (boohoo, special case). The repeating structure of the bottle caps, while not what you may initially think, is not too hard to figure out.

3.2 Algebraic Solution

Let $\vec{i}$ be the initial state, $\vec{f}$ be the final state, c be the cap-based transactions, and b be the bottle-based transaction.

$$\begin{aligned}
\vec{i} + c\begin{bmatrix}-3\\1\end{bmatrix} + b\begin{bmatrix}1\\-1\end{bmatrix} &= \vec{f}\\
c\begin{bmatrix}-3\\1\end{bmatrix} + b\begin{bmatrix}1\\-1\end{bmatrix} &= \vec{f} - \vec{i}\\
\begin{bmatrix}-3 & 1\\1 & -1\end{bmatrix}\begin{bmatrix}c\\b\end{bmatrix} &= \vec{f} - \vec{i}\\
\begin{bmatrix}c\\b\end{bmatrix} &= \frac{1}{2}\begin{bmatrix}-1 & -1\\-1 & -3\end{bmatrix}(\vec{f} - \vec{i})
\end{aligned}$$

The goal is to get as drunk as possible, which is reaching as many transactions as possible, thus we want to maximize $b + c$ with respect to the elements of f.

$$c + b = \frac{1}{2}(-2(f_c - i_c) - 4(f_b - i_b))$$

Since there's not that many possible combinations of $\vec{f}$, it would be practical in this case to simply check for the maximum, but that's no fun.

4 Using the final state of the vector to deduce the amount of drinks drunk

Since two empty bottles can get you a drink and a drink is worth \$2, then that means one bottle is worth \$1. Similarly since four bottle caps can get you a drink and a drink is worth \$2, then that means bottle cap is worth \$0.5.

Since a full drink is consisted of one cap, one bottle, and some drink, we can use simple algebra to deduce that:

$$\$2 = d + \$1 + \$0.5 \tag{1}$$

$$d = \$0.5 \tag{2}$$

the worth of the drink is \$0.5. If we started with \$10 dollars, and we are left with a bottles and b caps, then:

$$\$10 = \$0.5x + \$a + \$0.5b \tag{3}$$

$$x = \frac{\$10 - \$a - \$0.5b}{\$0.5} \tag{4}$$